チュートリアル: α クオーツを例にして

平成 28 年 9 月 4 日

1 はじめに

このチュートリアルでは α クオーツの構造最適化、projected DOS の計算、およびバンド構造の計算、セル構造の最適化を行う。

このチュートリアルでは、Intel Fortran Compiler と OpenMPI を使っていることを前提にする。そうではない場合には、論理機番の設定方法や mpirun の起動方法に読み替えが必要であるが、gfortran については同じ方法で論理機番が設定できるように xTAPP 自体が対応している。

なおこの配布パッケージにはチュートリアルの入力例と出力例が同梱されている。`sample/tutorial` を見ること。

2 α クオーツの構造

- 空間群 $P3_221$ No. 154
- $a = 4.9134$ Å
- $c = 5.4052$ Å
- $\alpha = \beta = 90°$
- $\gamma = 120°$
- 一セル中の SiO_2 の数 $Z = 3$
- 原点のオフセット $(0, 0, -1/3)$
- 原子位置（セル座標）

```
Si 3a 0.4699 0.0000 0.0000
O  6c 0.4141 0.2681 0.1188
```

2 列目は Wyckoff position の記号である。

Wyckoff position の計算は、bilbao crystallographic server, `http://www.cryst.ehu.es/`で行ってくれる。

Si の座標から origin のシフト量が $(0, 0, 2/3)$ として、3a は

```
(x,0,2/3)
(0,x,1/3)
(-x,-x,0)
```

であり、6c は

```
(x,y,z)
(-y,x-y,z+2/3)
(-x+y,-x,z+1/3)
(y,x,-z)
(x-y,-y,-z+1/3)
(-x,-x+y,-z+2/3)
```

である。

これら Wyckoff position を原点のオフセット分ずらしておいて全部の原子の位置がセル座標で

```
Si  0.4699  0.0000  0.666666666666
Si  0.0000  0.4699  0.333333333333
Si -0.4699 -0.4699  0.0
O   0.4141  0.2681  0.785466666666
O  -0.2681  0.1460  0.452133333333
O  -0.1460 -0.4141  0.1188
O   0.2681  0.4141  0.214533333333
O   0.1460 -0.2681  0.547866666666
O  -0.4141 -0.1460  0.8812
```

とわかる。

同じサイトで General position を調べればそれは実空間での対称性の情報となる。なお、実際にはこれは 6c と同じである。

これを使って「逆空間で」働く対称性行列は

```
 1  0  0   0
 0  1  0   0
 0  0  1   0

-1 -1  0   0
 1  0  0   0
 0  0  1   2/3

 0  1  0   0
-1 -1  0   0
 0  0  1   1/3

 0  1  0   0
 1  0  0   0
 0  0 -1   0

 1  0  0   0
-1 -1  0   0
 0  0 -1   1/3

-1 -1  0   0
 0  1  0   0
 0  0 -1   2/3
```

である。反転対称性はない。この際に必要な行列計算は octave か MATLAB を使うとできる。

この計算は、逆空間での対称操作の行列要素は実空間の行列の逆行列の転置行列であることと、General position が primitive cell ではなく、conventional cell について書かれていることを知っていればできるが、実際にこの計算を行い、その結果を xTAPP の入力フォーマットが必要とする順番で出力する octave のプログラムがサンプルの `symop-quartz.m` である。

これを使用する時には、あらかじめ表示フォーマットを rational approximation を使用するようにセットしておくと良い。

```
octave:1> format rat
octave:2> source symop-quartz.m
```

なお対称操作を表現する `symopr` 関数と primitive cell を conventional cell の基本ベクトルで表現する行列 `aa` を適宜書き換えることで、その他の対称性についても容易に計算できる。

3 原子位置から対称性を見つける方法

このチュートリアルでは空間群の種類が既知であることを前提にしている。決まった結晶構造から計算を始める場合にはこの前提が正しいはずである。しかしながら空間群が未知で原子の位置のみが既知である場合には、空間群を決めなくてはならない。

3.1 FINDSYM というインターネットのサイトを使う方法

サイトは `http://stokes.byu.edu/iso/findsym.php` である。

このサイトに先のデータを入力すると空間群の番号、Wyckoff position、対称性行列を含む cif ファイルが得られるのでここから始めれば良い。

なお、`Type of each atom in the unit cell` には、この例の場合 `3*Si 6*O` を与える。

`Position of each atom in the unit cell` には

```
 0.4699  0.0000  0.666666666666
 0.0000  0.4699  0.333333333333
-0.4699 -0.4699  0.0
 0.4141  0.2681  0.785466666666
-0.2681  0.1460  0.452133333333
-0.1460 -0.4141  0.1188
 0.2681  0.4141  0.214533333333
 0.1460 -0.2681  0.547866666666
-0.4141 -0.1460  0.8812
```

を与える。

3.2 phonopy の対称性機能を使う方法

phonopy には原子構造の対称性を発見する機能がある。これを使って対称性データをつくるには、まず原子構造を POSCAR ファイルに書き込み、それを phonopy に解析させ、その結果を xTAPP 向けに変換すれば良い。

POSCAR ファイルの生成には xTAPP 付属の xticonv コマンド phonopy の対称性データファイル (YAML 形式) の変換には xTAPP-util 付属の sym2xti.py を用いる。

xTAPP の入力ファイル quartz.cg から必要な対称性データを xTAPP の入力形式で生成してファイル symdata に格納する手順の例は以下のようになる。

```
$ xticonv poscar quartz.cg > POSCAR
$ phonopy --symmetry > sym.yml
$ sym2xti.py sym.yml > symdata
```

4 入力データの用意

入力データは

```
# [名前]
...
```

の形式を取っているセクションに分かれているテキストファイルである。セクションには Fortran の namelist が含まれていることがある。以下順番にそれぞれのセクションを書いていく。セクションの順番には意味がないのでどの順番に書いても良い。詳しくは inputformat.tex を参照すること。

以下の例では namelist 中に文字列を指定している部分があるが、この文字列を必ず引用符でくくらないとならない処理系があるので注意をすること。gfortran はそのような例である。

なおこの節で作成する各セクションはまとめて quartz.cg に収納する。

4.1 原子構造データの構成

対称性を指定する symmetry data セクションは

```
# symmetry data
&symmetry
symmetry_format = 'reciprocal',
number_sym_op = 6,
has_inversion = 0,
denom_trans = 3
/
 1  0  0  0  1  0  0  0  1  0  0  0
-1  1  0 -1  0  0  0  0  1  0  0  2
 0 -1  0  1 -1  0  0  0  1  0  0  1
 0  1  0  1  0  0  0  0 -1  0  0  0
 1 -1  0  0 -1  0  0  0 -1  0  0  1
-1  0  0 -1  1  0  0  0 -1  0  0  2
```

となる。原子の位置情報などを指定する atom data セクションは、使用する擬ポテンシャルは 4 価のシリコンと 6 価の酸素としそれぞれ 1 番目と 2 番目の擬ポテンシャルとすると

```
# atom data
 4  14
 6  8
 1   0.4699  0.0000  0.666666666666
 1   0.0000  0.4699  0.333333333333
 1  -0.4699 -0.4699  0.0
 2   0.4141  0.2681  0.785466666666
 2  -0.2681  0.1460  0.452133333333
 2  -0.1460 -0.4141  0.1188
 2   0.2681  0.4141  0.214533333333
 2   0.1460 -0.2681  0.547866666666
 2  -0.4141 -0.1460  0.8812
```

となる。ここで 2 ,3 行目はそれぞれシリコンの価電価が 4 、原子番号が 14 であり、酸素の価電荷が 6、原子番号が 8 であることを指定している。この情報はそれぞれ 1,2 番目の擬ポテンシャルと照合される。

4.2 サンプル k 点データの構成

$6 \times 6 \times 4$ のサンプル k 点を構成する。Γ 点を通らないメッシュを作る。サンプル k 点関連の情報を指定する k-points data セクションは

```
# k-points data
&smpl_kpt
dos_mode = 'COS',
dos_mesh = 2, 2, 2,
bz_mesh = 12,
bz_number_tile = 1
/
 6  6  6
 2  2  2
```

となる。これは

1. フェルミ面積分は cos 展開法を使う。
2. cos 展開したバンド分散は $2 \times 2 \times 2$ のメッシュで積分する。
3. サンプル k 点の位置を指定するため第一ブリュアンゾーンを $12 \times 12 \times 12$ に等分割する。分割でできる格子は原点を通る。以下この格子を拡張ゾーンに拡大して考える。
4. 格子点の (6,6,6) の位置から始めて、(2,2,2) の間隔でサンプル k 点を格子状に選出する。a, b, c 軸方向のサンプル位置は第一ブリュアンゾーンに限定すると

   a, b, c: 6, 4, 2, 0, -2, -4, (-6)

   となる。なおブリュアンゾーンの端は周期的につながっていることに注意すること。

5. $12/2 \times 12/2 \times 12/2 = 6 \times 6 \times 6$ のサンプル k 点が得られる。

6. このサンプル k 点に対称性でつながっているすべての k 点を集めたものが全サンプル k 点である。この場合には、元のサンプル k 点と同一の集合であるが、サンプリングの原点の位置によってはこの数が $6 \times 6 \times 6$ よりも大きいことがある。

を意味する。ここで α クオーツは絶縁体なので、バンドの分散関係 $\epsilon_n(k)$ の cos 展開は実質必要でないため `dos_mesh` は適度に小さいメッシュを取っている。

4.3 計算条件のセットアップ

平面波基底のカットオフエネルギーを 36Ry に取るには、波数を 6 に取れば良い。また、スピンなしの計算を行うこととする。その他の条件を、

- 格子定数 $a = 9.2850$ bohr であり、$c/a = 1.1001$ である
- 元素の数は 2 個。原子数は全部で 9 個。
- 価電子数は 48 個である。したがって価電子帯に必要なバンドは 24 本。計算するバンド数は少し多めに 28 本とする。
- SCF は 30 回も回せばこの場合収束するはずである。
- 交換相関汎関数に PBE を使う。
- 多めに計算の打ち切り時間を 7200 秒にセット。
- 終了時に波動関数を書き出しておく。

とすると、`main data` セクションは

```
# main data
&tappinput
lattice_factor = 9.2850,
lattice_list = 1.0, 0.0,  0.0, -0.5,  0.866025403784439,  0.0,  0.0,  0.0,   1.1001,
cutoff_wave_function = 6.0,
number_element = 2,
number_atom = 9,
number_band = 28,
store_wfn = 1,
scf_number_iter_1st = 30,
scf_number_iter = 30,
xc_type = 'PBE',
control_uptime = 7200.0
/
```

とする。

ここで、`lattice_factor` は計算セル＝単位格子の典型的な長さ、`lattice_list` は計算セルの基本並進ベクトル $\mathbf{a}_1$、$\mathbf{a}_2$、$\mathbf{a}_3$ について $\mathbf{a}_1$ から順にそれぞれの x, y, z 成分を並べたもので、その単位は `lattice_factor` 単位である。

構造最適化を行うので、さらに struct_opt data セクションをセットしておく。構造最適化を力場が 1×10^{-4}hartree/bohr まで小さくなるように行うには

```
# struct_opt data
&struct_opt
converge_force = 1.0d-4
/
```

と設定する。また特に制約を課さずに最適化するので、構造最適化時の制約条件を指定する str_opt_constr data セクションを

```
# str_opt_constr data
 1
 0
```

としておく。これは最適化のための固定行列が自明な 1 個だけであり、それ以外に設定される原子の個数が 0 であることを示している。

4.4 入力ファイルの確認

quartz.cg に入っている構造データを可視化のために変換するツール xticonv が xTAPP の配布キットに入っている。これを用いて CIF 形式 (http://www.iucr.org/resources/cif) に変換するためには

```
$ xticonv cif quartz.cg > quartz.cif
```

とする。XYZ 形式なら

```
$ xticonv xyz quartz.cg > quartz.xyz
```

である。

CIF 形式や XYZ 形式のファイルの可視化は、VESTA、Jmol などでできる。

4.5 入出力ファイルの一括指定

xTAPP はいくつものファイルを作成するが、この文書ではそれらの名前を与えた基幹部分と既定のサフィックスの形で指定することにする。この場合には file map data セクションを設定するのが良い。基幹部分を quartz とし、使用する擬ポテンシャルとしてカレントディレクトリにある ps-Si-pbe と ps-O-pbe をそれぞれ 1 番目、2 番目の原子種に使用することにした場合、

```
# file map data
&filemap
basename = quartz,
number_PP_file = 2
/
ps-Si-pbe
ps-O-pbe
```

とする。

5　構造最適化プログラムの実行

擬ポテンシャルは xTAPP-test に付属の ps-Si-pbe と ps-O-pbe を用いる。これらは PS ディレクトリ以下にある。これらをカレントディレクトリにコピーしておく。

まず、初期化プログラム inipot を動かす。プログラムはカレントディレクトリにコピーしてあるものとする。同様な構造で一度動かせば良く、毎回実行する必要はない。

inipot は作った入力ファイルを論理機番 10 から読み込む。sh 系での入力コマンドは

```
$ export FORT10=./quartz.cg
$ mpirun ./inipot > inipot.log
```

である。このコマンドで 1 MPI process をつかって inipot が動く。ログは inipot.log に回収している。実行が正常に終わるとカレントディレクトリにいくつかの中間ファイルが作成される。

次に構造最適化を行うプログラム cgmrpt を動かす。プログラムはカレントディレクトリにコピーしてあるものとする。mpirun に渡すランクファイルを 6corefproc.rkf としておく。これの例は xTAPP-test にあるが本質的に実行環境依存である。例えばバッチシステムが動いている場合には必要ないはずである。

2 MPI process で計算を実行させる入力コマンドは

```
$ export FORT10=./quartz.cg
$ mpirun -np 2 -rf 6corefproc.rkf ./cgmrpt > cgmrpt.log
```

である。このコマンドで 2 MPI process をつかって cgmrpt が動く。ログは cgmrpt.log に回収している。

この実行においては波動関数、ローカルポテンシャル、電荷密度、結果のサマリがそれぞれ

```
quartz.wfn
quartz.lpt
quartz.rho
quartz.str
```

に保存される。これらサフィックスの詳細は”programs.tex” を参照すること。

6 結果の確認

6.1 ログファイルの見方

構造最適化がどのように収束したかは、ログを MAX FORCE で検索すれば良い。MAX FORCE は原子にかかっている力の最大を示している。収束していなくても計算は一定回数で打ちきられるため、必ず MAX FORCE を確認しておくこと。

構造最適化で得られた原子構造、その全エネルギーなどのデータは quartz.str に出力されている。

- lattice_factor と lattice_list は入力ファイルと同じ意味。
- total_energy には全エネルギー E
- stress_tensor にはストレスのテンソル
$$-\frac{\partial E}{\partial \epsilon_\nu}; \quad \nu \text{は } xx, yy, zz, yz, zx, xy \text{ の順}$$
- fermi_energy はフェルミエネルギー。それぞれのスピン成分ごとに個数を決めるモードがあるので二つある。
- # atom_kind, atom_position by cell coordinate 以下に原子種の番号とセル座標での原子位置
- # force by Cartesian coordinate 以下に原子の力場。原子単位系 (=hartree/bohr)、デカルト座標系での力。

同様の出力はログにもでている。

- TOTAL ENERGY に全エネルギー
- TOTAL STRESS に全エネルギーの体積微分
- TOTAL FORCE IN HT に力場のデカルト座標成分
- FERMI ENERGY にフェルミエネルギー
- UNIT CELL にセル形状

などである。

6.2 最適化構造の可視化

quartz.str に入っている構造データを可視化のために変換するツール strconv が xTAPP の配布キットに入っている。これを用いて CIF 形式 (http://www.iucr.org/resources/cif) に変換するためには

```
$ strconv cif quartz.str > quartz.cif
```

とする。XYZ 形式なら

```
$ strconv xyz quartz.str > quartz.xyz
```

である。

CIF 形式や XYZ 形式のファイルの可視化は、VESTA、Jmol などでできる。

7　バンド図を描く

7.1　トレースする k 点の決定

バンド図を描くために必要な特殊 k 点の位置も、bilbao crystallographic server、`http://www.cryst.ehu.es/` で行ってくれる。これらのつながり具合を Brillouin Zone で確認することもここでできる。

トレースする k 点を順に

1. G (0,0,0)
2. K (1/3, 1/3, 0)
3. H (1/3, 1/3, 1/2)
4. A (0,0,1/2)
5. G (0,0,0)
6. M (1/2,0,0)

と設定し、それぞれの間の分点を

1. GK 10
2. KH 10
3. HA 10
4. AG 10
5. GM 10

と設定する。

7.2　入力データの準備

次にバンドを計算するプログラム `vbpef` が必要とするセクション `trace band data` を作成し、構造最適化に使った入力データに付け加える。また計算するバンド数はシリコン由来の非占有バンドも含めて 36 本とする。

作成するデータは 5 区間分であるから

```
# trace band data
 &trace_band
 number_band = 36,
 number_trace_block = 5
 /
   'G'          'K'          'H'          'A'          'G'          'M'
   0.00000000   0.33333333   0.33333333   0.00000000   0.00000000   0.50000000
   0.00000000   0.33333333   0.33333333   0.00000000   0.00000000   0.00000000
   0.00000000   0.00000000   0.50000000   0.50000000   0.00000000   0.00000000
              10           10           10           10           10
```

である。各列に k 点のシンボルと位置を指定している。最後の列は区間毎の分割数である。これをもとの quartz.cg に付け加えて quartz.pef としておく。

次に `main data` セクションにおいて、namelist の`&tappinput` に `initial_lpt=1` を付け加えて、`vbpef` が構造最適化時に計算済みのローカルポテンシャルを読み込むように設定しておく。

そして、構造最適化済みの構造データで quartz.pef の `atom data` セクションを置き換えておく。

7.3 バンド計算プログラムの実行

擬ポテンシャルは xTAPP-test に付属の ps-Si-pbe と ps-O-pbe を用いる。これらは PS ディレクトリ以下にある。

まず、初期化プログラム `inipot` を動かす。プログラムはカレントディレクトリにコピーしてあるものとする。同様な構造で一度動かせば良く、毎回実行する必要はない。

`inipot` は作った入力ファイルを論理機番 10 から読み込む。sh 系での入力コマンドは

```
$ export FORT10=./quartz.pef
$ mpirun ./inipot > inipot.log
```

である。このコマンドで 1 MPI process をつかって `inipot` が動く。ログは inipot.log に回収している。

次に、バンド計算プログラム `vbpef` を動かす。プログラムはカレントディレクトリにコピーしてあるものとする。このプログラムは収束済みのローカルポテンシャルを必要とするので注意すること。また、このローカルポテンシャルを計算したときの原子構造と同じ原子構造を `vbpef` に渡す必要がある。この原子構造データを入力ファイルからではなく構造最適化が収束した時の中間ファイルから引き継がせると手間が省けるが、このためには `main data` セクションで namelist の `tapp input` で `chain_calc = 3` を設定すれば良い。

`vbpef` の追跡した k 点の固有値データを `quartz.band` に格納させることにすると、sh 系での入力コマンドは

```
$ export FORT10=./quartz.pef
$ mpirun -np 2 -rf 6corefproc.rkf ./vbpef > vbpef.log
```

である。このコマンドで 2 MPI process をつかって `vbpef` が動く。ログは vbpef.log に回収している。

7.4 バンド図を作成する

諏訪氏が開発したものを多少改変したバンド図作成ツール vbpef2gp-lsda によって、追跡した k 点の固有値データからバンド図を描くことができる。

使用方法は

```
$ vbpef2gp-lsda -fquartz ./quartz.band
```

である。これでカレントディレクトリにある quartz.band が処理されて quartz.gp、quartz.dat、quartz.kpt が作成される。次に gnuplot で quartz.gp を処理するとバンド図が gnuplot で作成される。

8 projected DOS を計算する

酸素の s 軌道、p 軌道への projected DOS を計算する。すでに計算した波動関数が必要である。

8.1 入力データの用意

酸素の軌道のデータは擬ポテンシャル生成プログラムの一つ solps で作成する。擬ポテンシャルに必要な軌道データが付属していない場合、改めて入手する必要がある。ここでは xTAPP-test に付属の quartz-wfn2chg.pwav.s と quartz-wfn2chg.pwav.p を使う。それぞれ s 軌道と p 軌道である。

次に quartz.pef をコピーして quartz.w2c とし、以下の編集を行う。まず、セクション+`main data` の namelist の`&tappinput` において

```
initial_wfn = 1
```

を付け加えて、波動関数を読み込むようにする。次に、新しいセクション `inspect wfn data` を追加する。

```
# inspect wfn data
 &inspect_wfn
 distrib_mode = 'pdos',
 pdos_target_atom = 4
 /
```

ここでは 4 番目の原子の位置の projected DOS を計算することにする。

8.2 projected DOS の計算プログラムの実行

プログラム `wfn2chg` を動かす。プログラムはカレントディレクトリにコピーしてあるものとする。このプログラムは収束済みのローカルポテンシャルを必要とするので注意すること。また、このローカルポテンシャルを計算したときの原子構造と同じ原子構造を `wfn2chg` に渡すこと。この原子構造データを入力ファイルからではなく構造最適化が収束した時の中間ファイルから引き継がせると手間が省けるが、このためには `main data` セクションで namelist の `tapp input` で `chain_calc = 3` を設定すれば良い。

計算は各軌道ごとに別に行う。ここでは s 軌道について説明するが p 軌道でも同様である。

入出力ファイルとして

1. 軌道データとして `quartz-wfn2chg.pwav.s`
2. `wfn2chg` の計算したデータを `quartz.dosms.s` に格納させる

3. 入力波動関数は quartz.wfn

を仮定するが、プログラムは基幹名を使って 1 を quartz.pwav として入力し、2 を quartz.dosms として出力するので、プログラムの前後でコピーと名称変更を行なう。

sh 系での入力コマンドは

```
$ export FORT18=./quartz-wfn2chg.pwav.s FORT95=./quartz.wfn
$ cp quartz-wfn2chg.pwav.s quartz.pwav
$ export FORT10=./quartz.w2c
$ mpirun ./wfn2chg > wfn2chg.s.log
$ mv quartz.dosms quartz.dosms.s
```

である。ログは wfn2chg.s.log に回収している。

次に xTAPP 本体に付属のユーティリティプログラム tetrapdos を用いて quartz.dosms.s から projected DOS を四面体法で計算する。tetrapdos はあらかじめカレントディレクトリにコピーしてあるものとする。

projected DOS の計算は計算上の最大軌道エネルギーから最小軌道エネルギーまで 300 点のメッシュで行なうことにすると sh 系での入力コマンドは

```
$ ./tetrapdos quartz.dosms.s 300 1 > quartz.dosms.s.pdos
```

である。すると、quartz.dosms.s.pdos が作成されて、ここに s 軌道の projected DOS のデータが記録される。出力のフォーマットは programs.tex を参照すること。

9 セル形状の最適化を行う

外部圧力が 10 GPa の下でセル形状を最適化する。

9.1 入力データの準備

quartz.cg を quartz-cell.cg にコピーして、struct_opt data セクションを以下のように変更する。

```
# struct_opt data
&struct_opt
converge_force = 1.0d-3,
converge_stress = 1.0d-3,
stress_scale = 1.0, 1.0, 1.0, 1.0, 1.0, 1.0,
extern_pressure = 0.00033989313487924 89d0
/
```

この様にすることで原子位置に加えてセル形状も最適化されるようになる。ここで新規に設定した stress_scale は、セルに働くストレスを原子に働く力場相当に変換する係数である。外部圧力は

extern_pressure に原子単位系で指定する。なお指定していなければ 0 となる。converge_stress は、セルのストレスと外部圧力の釣り合いの収束を指定するもので、

$$\frac{1}{N_{atom}}\left(\frac{dE}{d\epsilon_\nu} - p_{ext}V\right); \quad \text{for} \quad \nu = xx, yy, zz$$

$$\frac{1}{N_{atom}}\left(\frac{dE}{d\epsilon_\nu}\right); \quad \text{for} \quad \nu = yz, zx, xy$$

の許容最大を指定する。

またセル形状が変化する場合には、Bernasconi-Tosatti-Parrinello による運動エネルギーの補正を使う必要がある。ここでは main data セクションの namelist の&tappinput を以下のように変更する。

```
cutoff_wave_function = 8.0,
cutoff_btp_a = 60.0,
cutoff_btp_gc = 7.0,
cutoff_btp_sigma = 1.0
```

これらのパラメータはセル変形に対してストレスが精度良くなめらかに計算されるように選ばなければならない。さもないと構造最適化が収束しなくなる。小さな converge_force を使うためにはより厳しい条件、つまりより多くの平面波基底が必要である。

最後にセクション file map data の基幹名の同様に quartz-cell に変更しておく。

9.2 構造最適化の実行

通常の構造最適化と同様に実行させる。平面波カットオフなど条件を変えているのでもう一度 inipot を実行する必要がある。

うまく計算できている場合、cgmrpt のログの XMIN PAR:の行の最後にある IEX:の値に 1 はあまりでてこないはずである。